Performance Evaluation for Electromagnetic Solvers with Interpolatory and Hierarchical Nédélec Vector Bases

Jin Xu,[1,*] Xiaohe Zhufu,[1] Ruifeng Zhao,[1] Jiachang Sun,[1] Chao Li,[2] Lei Yang,[2] and Misun Min[3]

[1] *Institute of Software, Chinese Academy of Science*
#4 Zhong Guancun South 4th Street, 100190, Beijng, China
[2] *Institute of Modern Physics, Chinese Academy of Science*
#509 Nanchang Road, 730000, Lanzhou, China
[3] *Mathematics and Computer Science Division*
Argonne National Laboratory
9700 S. Cass Ave., Argonne, IL 60439, USA.

Abstract. Since Nédélec published his milestone papers in the 1980s, edge vector bases have been widely used in electromagnetic simulations. They have been proved to be more suitable than nodal bases for simulating electric and magnetic fields by satisfying edge continuity. Many different types of Nédélec vector bases have been developed. These bases can be classified into two main categories: interpolatory and hierachical types. The hierarchical vector bases can use either uniform or nonuniform meshes, while the interpolatory vector bases can use only uniform meshes. These bases have been designed by different researchers in the past two decades, but their performances have not been compared in detail before. Interpolatory vector basis are relatively easy to implement, while hierarchical vector bases can use multigrid techniques for speedup in solving linear systems. This paper studies these two vector bases through detailed numerical comparisons. Our objective is to enable researchers to choose better vector bases for various applications and to develop new vector bases in the future.

AMS subject classifications: 65C20, 65Z05, 68U20, 68W10
Key words: Eigenvalue Problems, Maxwell's Equation, *hp*-Finite Element Method, Nédélec Basis, Lanczos Algorithm

1 Introduction

Electromagnetic (EM) phenomena arise in many fields of science and engineering, such as radar, wireless communication, accelerators, and electric circuits. EM simulations usu-

*Corresponding author. *Email addresses:* xu_jin@iscas.ac.cn (J. Xu), xiaohe@iscas.ac.cn (X. Zhufu), ruifeng@iscas.ac.cn (R. Zhao), sun@mail.rdcps.ac.cn (J, Sun), chli@impcas.ac.cn (Ch. Li), lyang@impcas.ac.cn (L. Yang), mmin@mcs.anl.gov (M. Min).

ally include time domain and frequency domain simulations. Eigenvalue problems are usually solved in the frequency domain. Traditional numerical approaches are finite-difference and finite-volume methods. Because of complicated geometry, however, the finite-element method (FEM) has proved most advantageous [15, 31, 33] and has been broadly used in EM simulations. Since high-order bases can dramatically increase the accuracy, the high-order *hp*-finite element method (*hp*-FEM) is preferred in the simulations. Such methods for solving Maxwell's equation are successful except when solving eigenvalue problems. The main reason is the observed appearance of spurious, nonphysical solutions when a straightforward nodal basis with continuous Galerkin (CG) FEM is used to discretize the Maxwell curl-curl equations. These problems may come from the poor representation of the large null space of the operator, which has troubled people for many years. Detailed discussion can be found in [32]. Although applying different weak forms of Maxwell's equation may overcome this difficulty with traditional nodal-type finite-element methods, the use of special curl-conforming elements [26, 27] removes the problem of spurious modes. FEMs based on edge elements, or Nédélec elements, have rapidly become the dominant approach for solving geometrically complex problems in the frequency domain.

The Nédélec vector bases belong to the curl-conforming space $H(curl) = \{\vec{v} \in L_2(\Omega)$, and $\nabla \times \vec{v} \in L_2(\Omega)\}$, whereas traditional vector bases belong to $(L_2(\Omega))^3$. The lowest-order Nédélec vector bases are often referred to as *edge bases* [3, 25, 36] or *Whitney forms* [37]. The search for a convenient set of basis functions for higher-order approximation has attracted a great deal of attention. Several variants of second-order basis functions have been described in [6, 10, 16, 19], while third-order elements have been considered in [4, 17, 20, 25]. The construction of elements of higher than third-order was considered by Graglia *et al.* [11], and a nodal basis was described by Webb [36] where basis functions of arbitrary order are presented based on the degrees of freedom outlined in [27]. As scalar elements, there are two types of high-order vector bases. The first type uses interpolatory basis function [1, 6, 11, 12, 16, 19, 29, 32, 34, 38], and the second type uses hierarchical basis functions [2, 35, 36, 39]. Both types span the same vector space and start from the zeroth-order basis functions. Their major difference is in their construction. The interpolatory basis functions are defined on a set of points on the element, such that each basis function vanishes at all the points except for one point. These basis functions have several advantages. Their coefficients have a physical interpretation as components of the vector field at the interpolatory points. Since they interpolate the tangential components of the vector field at the interpolatory points, their use makes it easy to enhance boundary conditions. Moreover, they have a unified expression, which significantly simplifies the implementation of solvers for generation of arbitrary-order basis functions. However, interpolatory basis functions of a given order are all different from those of the lower orders. Hence, different-order basis functions cannot be used together, these making it impossible to implement P-adaption. In contrast, the hierarchical basis functions are not defined on a set of points. Higher-order hierarchical basis functions are formed by adding new functions to the lower-order basis functions. Thus, the first-order basis

functions contain those of the zeroth order, the second-order function contains those of the zeroth and the first orders, and so on. The distinct advantage of these basis functions is that they permit the use of different orders in a problem and hence can be employed for P-adaption.

In this paper, we study the performance of these two types of vector bases for eigenvalue problems. The paper is organized in the following way: Maxwell's equations are given in Section 2, and the numerical methods for bases are introduced in Section 3; details are included in Appendixes A and B. The numerical and parallel methods for eigenvalue problems are discussed in Section 4, and verification and benchmarks are presented in Section 5. Applications in simple geometries and radio frequency (RF) resonators are given in Section 6. Our conclusions are summarized in Section 7.

2 Maxwell's Equations

Consider Maxwell's equations in a three-dimensional domain Ω with in a vacuum

$$\frac{\partial \vec{D}}{\partial t} = \nabla \times \vec{H} - \vec{J} \quad , \qquad \nabla \cdot \vec{D} = \rho, \tag{2.1}$$

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E} \quad , \qquad \nabla \cdot \vec{B} = 0, \tag{2.2}$$

where $\rho(\vec{x},t)$ is the charge distribution and the current $\vec{J}(\vec{x},t)$ is related to the electric field $\vec{E}(\vec{x},t)$ through Ohms law, $\vec{J} = \sigma \vec{E}$, where σ is the finite conductivity. The electric field $\vec{E}(\vec{x},t)$ and the electric displacement $\vec{D}(\vec{x},t)$, the magnetic field $\vec{B}(\vec{x},t)$, and the magnetizing field $\vec{H}(\vec{x},t)$ have following constitutive relations.

$$\vec{D} = \epsilon \vec{E} = \epsilon_0 \epsilon_r \vec{E}, \qquad \vec{B} = \mu \vec{H} = \mu_0 \mu_r \vec{H} \tag{2.3}$$

Here ϵ_r and μ_r are the relative permittivity and permeability of the material, respectively. For our simulations of vacuum, they both equal one; that is, the permittivity and permeability ϵ_0 and μ_0 satisfy $c^2 = 1/(\epsilon_0 \mu_0)$.

We first transform the fields into Fourier space.

$$\vec{\tilde{E}}(\vec{x},t) = \int_{-\infty}^{\infty} \vec{E}(\vec{x},t) e^{\omega t} d\omega, \qquad \vec{\tilde{H}}(\vec{x},t) = \int_{-\infty}^{\infty} \vec{H}(\vec{x},t) e^{\omega t} d\omega \tag{2.4}$$

Then Maxwell's equations can be written in frequency domain as follows.

$$\omega \epsilon \vec{\tilde{E}} = \nabla \times \vec{\tilde{H}} \quad , \qquad \nabla \cdot \epsilon \vec{\tilde{E}} = 0, \tag{2.5}$$

$$\omega \mu \vec{\tilde{H}} = -\nabla \times \vec{\tilde{E}} \quad , \qquad \nabla \cdot \mu \vec{\tilde{H}} = 0, \tag{2.6}$$

Figure 1: Interpolatory points (P=3) in a triangle (left) associated with edge-connecting nodes 1 and 2 and a tetrahedron (right) associated with edge-connecting nodes 2 and 3

Taking the cross-product operation over Equations (2.5)-(2.6), we obtain a second-order curl-curl form.

$$\nabla\times\nabla\times\vec{H}=\frac{\omega^2}{\bar{\epsilon}\bar{\mu}}\vec{H}=\frac{\omega^2}{c^2}\vec{H}=\lambda\vec{H} \quad , \qquad \nabla\cdot\vec{H}=0 \tag{2.7}$$

$$\nabla\times\nabla\times\vec{\vec{E}}=\frac{\omega^2}{\bar{\epsilon}\bar{\mu}}\vec{\vec{E}}=\frac{\omega^2}{c^2}\vec{\vec{E}}=\lambda\vec{\vec{E}} \quad , \qquad \nabla\cdot\vec{\vec{E}}=0 \tag{2.8}$$

For a perfectly electrically conducting (PEC) wall, the boundary conditions for the fields are as follows.

$$\vec{n}\times\vec{\vec{E}}=0, \qquad \vec{n}\cdot\vec{\vec{H}}=0 \tag{2.9}$$

3 Nédélec Vector Bases

In this section, we present two types of Nédélec vector bases. Nédélec vector bases in 2D on a triangle element are briefly explained first, followed by the vector bases on a 3D tetrahedron element. Different from the nodal FEM, the degrees of freedom for the Nédélec vector bases are associated with the edges and face in 2D, and with the edges, faces, and volume in 3D.

3.1 Interpolatory Nédélec Vector Bases

Our interpolatory vector bases are based on the work of Graglia *et al.* [11,12] and Jianmin *et al.* [15].IWe use Lagrange functions on uniform grid points. The lowest-order basis, which is the zeroth-order vector basis, is associated with edges. The high-order vector bases are constructed by multiplying a complete interpolatory polynomial of order P

Figure 2: First-order (P=1) interpolatory type Nédélec vector bases on a triangle element

with the zeroth-order bases. The left figure in Fig. 1 shows the interpolatory points for P=3 on a triangle associated with edge-connecting nodes 1 and 2, and the right figure is the interpolatory points for P=3 on a tetrahedron associated with edge-connecting nodes 2 and 3. Figure 2 shows the first-order interpolatory Nédélec vector bases on a triangle element. Details of the construction method can be found in Appendix A.

3.2 Hierachical Nédélec Vector Basis

Our hierarchical vector bases are based on the work of Webb *et al.* [35,36], Andersen and Volakis [2], and Zhu and Cangellaris [39]. The grid points on triangle and teterahedron can be either uniform or nonuniform. The lowest order is same as the interpolatory vector bases, which is the zeroth-order vector bases associated with edges. The (P+1)th-order vector bases contain all Pth-order vector bases. Figure 3 shows the first-order hierarchical type Nédélec vector bases on a triangle. Their shapes are different from those of the interpolatory vector bases shown in Fig. 2. This difference affects the numerical computations studied in this paper. Details of the construction method can be found in Appendix B.

4 Numerical and Parallel Methods

In this section, we explain the numerical and parallel methods for the electromagnetic solvers based on the vector bases introduced above. Since these vector bases are more complicated than scalar bases, many challenges have to be solved, especially when the high-order bases are used.

Figure 3: First-order (P=1) hierarchical type Nédélec vector bases on a triangle

Figure 4: Transformation between reference and arbitrary tetrahedron element

4.1 Transformations

The vector bases introduced before are defined on a reference element. For real applications. they must be transformed to an arbitrary element in the computation domain. The transformations should follow rules that can sustain the property of the vector bases.

As shown in Fig. 4, for any tetrahedron element $K \in \tau_h$, there is a map $F_K : \hat{K} \to K$ such that $F_K(\hat{K}) = K$ and

$$F_K \hat{\vec{x}} = B_K \hat{\vec{x}} + \vec{b}_K, \tag{4.1}$$

where $\tau_h = \{K\}$ is a discretization of the computation domain and B_K and b_K are determined by the coordinates of four vertices of the tetrahedron element K. Suppose F_K transforms $\hat{p}(\hat{\vec{x}}) \in H^1(\hat{K})$ to $p(\vec{x}) \in H^1(K)$ by $p \circ F_K = \hat{p}$. Then $\nabla p(\vec{x}) = (dF_K)^{-T} \hat{\nabla} \hat{p}(\hat{\vec{x}}) = (B_K^T)^{-1} \hat{\nabla} \hat{p}(\hat{\vec{x}})$. Then, for $\hat{\vec{u}}(\hat{\vec{x}}) \in H^p(curl, \hat{K})$, there is $\vec{u}(\vec{x}) \in H^p(curl, K)$ such that $\vec{u}(\vec{x}) \circ F_K = (B_K^T)^{-1} \hat{\vec{u}}(\hat{\vec{x}})$, which means

$$\vec{u}(\vec{x}) = (B_K^T)^{-1}\hat{\vec{u}}(\hat{\vec{x}}). \tag{4.2}$$

4.2 Vector Bases Connectivity

The principal challenge in using vector bases is to satisfy the conforming properties on the element interfaces. We explain the methods for a 3D tetrahedron element. All methods are the same for a 2D triangle element.

4.2.1 Interpolatory Vector Bases

For interpolatory vector bases, the property of a constant component along the direction of the corresponding edge associated with the node has been used for setting up the modes matching on the interface. Only the bases associated with edges and faces need to be considered for node matching on the tetrahedron element. Because the order of the vertices on edges and faces may be different for two neighboring tetrahedron elements, the coefficients and directions of the vector bases on the two neighboring elements may be different even though they are related to the same nodes on the edges and faces.

Edge modes matching is easy because only two possibilities exist and are easy to be decided according to the order of the vertices. The global id of the vertices has been used to decide the direction of the edges on each tetrahedron element.

Face modes matching needs more effort and the order of the vertices on each triangle needs to be decided. Since there are two orders of the vertexes, clockwise and counter clockwise, the edge directions are opposite in these two cases. These need to be determined for each shared face on two adjacent tetrahedron elements. Variables have been set for edges and faces on each element in a preprocessing step in order to identify the types for these edges and faces shared by different tetrahedron elements. From the definition of the interpolatory vector bases, the construction of face vector bases can choose any two edges to multiply with a complete set of P-th order polynomial functions. For convenience, in our solvers we first decide which one in each two tetrahedron elements is the primary, and the other one is the secondary. Then the face vector bases on the primary element are chosen with the first two edges on the triangle. A function needs to be developed next in order to decide which two edges on the secondary element have been used to match the corresponding edges on the opposite primary tetrahedron element. Both the edge number and their directions need to be identified. Faces on the boundary are set to be the primary faces, and the first two edges are used to construct the vector bases. When all these information have been obtained, face modes can be exactly matched on shared faces for each two neighboring tetrahedron elements.

4.2.2 Hierarchical Vector Bases

For hierarchical vector bases, the mode matching is different from the case of interpolatory vector bases. The edge mode matching is similar as only two possibilities exist and they are easy to be decided according to the global ids of the vertices.

For face vector mode, since the bases are associated with vertexes, the correct orders of the vertexes should be decided. Since face mode matching is connected to the edge mode matching, the edge mode matching must be determined first. The face mode matching could be performed on the real tetrahedron element, or the vector field could be transformed to the standard tetrahedron element.

To transform the vector field with the hierarchical vector bases, we need to construct mass matrices for edges, faces, and volume. Their definition is given in the following.

For the edge mass matrix ME^k

$$ME_{ij}^k = \int_{edge} (\vec{\phi}_i^k \cdot \vec{e})(\vec{\phi}_j^k \cdot \vec{e}) dl, \tag{4.3}$$

where $\vec{e}$ is the unit vector along the edge k, and $\vec{\phi}_i^k$ is the I th vector base.

For the face mass matrix MF^k

$$MF_{ij}^k = \int_{face} Project(\vec{\phi}_i^k) Project(\vec{\phi}_j^k) ds, \tag{4.4}$$

where Project($\vec{\phi}_i$) is the projection of I th vector base on the face k.

For the volume mass matrix MV^k

$$MV_{ij}^k = \int_{volume} \vec{\phi}_i^k \cdot \vec{\phi}_j^k dv, \tag{4.5}$$

where $\vec{\phi}_i^k$ is the I th vector base on the tetrahedron element.

Besides these matrices, we need to compute correlated matrices for edge-face, edge-volume, and face-volume. They are defined in the following.

For the edge-face mass matrix MEF^{kl}

$$MEF_{ij}^{kl} = \int_{face} Project(\vec{\phi}_i^k) \cdot Project(\vec{\phi}_j^l) dv, \tag{4.6}$$

where Project($\vec{\phi}_i^k$) is projection of the I th vector base associated with edge k, and Project($\vec{\phi}_j^l$) is the projection of J th vector base associated with face l.

For the edge-volume mass matrix MEV^k

$$MEV_{ij}^k = \int_{volume} \vec{\phi}_i^k \cdot \vec{\phi}_j^k dv, \tag{4.7}$$

where $\vec{\phi}_i^k$ is the I th vector base associated with edge k, and $\vec{\phi}_j^k$ is the J th vector base associated with volume.

For the face-volume mass matrix MFV^k

$$MFV_{ij}^k = \int_{volume} \vec{\phi}_i^k \cdot \vec{\phi}_j^k dv, \tag{4.8}$$

where $\vec{\phi}_i^k$ is the I th vector base associated with face k, and $\vec{\phi}_j^k$ is the J th vector base associated with volume.

4.3 Eigenvalue Computation

From (2.7)-(2.8), we arrive at a general eigenvalue problem. We seek the eigenvalue λ_i and corresponding eigenvector $\vec{E}_i \in H^p(curl,\Omega)$, such that

$$A \cdot \vec{\tilde{E}}_i = \lambda_i M \cdot \vec{\tilde{E}}_i \tag{4.9}$$

$$A = (\nabla \times \vec{\tilde{E}}_i, \nabla \times \vec{\tilde{E}}_i) \tag{4.10}$$

$$B = (\vec{\tilde{E}}_i, \vec{\tilde{E}}_i). \tag{4.11}$$

Many different algorithms could be used for solving the general eigenvalue problem. If the problem size is not too large, direct methods such as LAPACK routines can be applied to extract the eigenvalues efficiently. In real applications, however, most problems have large sizes that make iterative methods more suitable. Among them, Krylov subspace projection methods are most popular. The basic idea of Krylov subspace projection methods is approximating the vector $\vec{v}$ by a vector $\tilde{v}$ that belongs to the expansion subspace K, by imposing the constraint that the residual vector $A\tilde{v} - \tilde{\lambda}\tilde{v}$ is orthogonal to the projection subspace L. Through this expansion, the original eigenvalue problem is projected onto an eigenvalue problem of smaller dimension. Then it is straightforward to compute the eigenvalues of this smaller system. Upon convergence, the eigenvalues of the original system can be easily computed. The most popular algorithms in this category are Arnoldi and Lanczos algorithms, which deal with general and symmetric matrixes, respectively. The publicly available software PARPACK [21,24] has been used to solve the eigenvalue problem on parallel supercomputers. Several approaches exist in PARPACK software to solve this general eigenvalue problem. The shift-and-invert mode has been used, which solves the following eigenvalue problem instead of the original one.

$$(A - \sigma M)^{-1} M \cdot \vec{\tilde{E}} = \mu \vec{\tilde{E}}, \qquad \mu = \frac{1}{\lambda - \sigma} \tag{4.12}$$

4.4 Mesh and I/O

We generate mesh using CUBIT software. With CUBIT software, several million tetrahedron elements can be generated on a desktop computer. Since vector bases have different format from that of the scalar bases, we have developed specific I/O routines to output data in the computation. Both eigenvalues and eigenvectors are stored for postprocessing. TECPLOT software has been used for visualizing the results.

4.5 Parallel Method

Because the eigenvalue computation may need a large mesh, parallel computation is necessary. For parallel processing, we have developed both global and local maps for the degrees of freedom. Interface maps have also been developed to match the mesh points on the interfaces. MPI functions have been used for communication between processors. We have developed several routines for the mesh partitioning. The simplest one evenly distributes the elements by its global id to all processors. A more meaningful approach is dividing the domain evenly according to the location in the domain. The publicly available software packages METIS and PARMETIS have also been used for mesh partition.

5 Verifications and Benchmarks

Since the constructions of the interpolatory and hierarchical vector bases rely on scalar bases, we investigate these scalar bases first. Since the cases in 2D are similar to those in 3D, we focus on 3D. We first study three types of scalar bases: Silvester nodal base [31], barycentric modal base, and nodal base built from orthogonal bases on a tetrahedron element [14]. Then we compare the vector bases for the 3D domain with a tetrahedron element. We also test these two types of vector bases in 3D for eigenvalue problems.

5.1 Scalar Bases Comparisons

Suppose the standard reference tetrahedron element has the definition of $[-1,1]^3$ and the global domain also is $[-1,1]^3$.

5.1.1 Interpolation

The interpolation errors for the scalar bases are shown on the left of Fig. 5. The error is defined as L_∞. The test function is $f(x,y,z) = exp(x+y+z)$, and the number of total tetrahedron elements is 204. The computation domain is $[-1,1]^3$. Since the Silvester nodal base is defined on a uniform mesh, the error is computed on uniform mesh. For a barycentric nodal base and a nodal base built from an orthogonal base, a nonuniform mesh has been used. Therefore, three lines have been shown on the left of Fig. 5. The nonuniform mesh uses Fekete points on the tetrahedron element. One can see that the accuracies of these three scalar bases are close: the barycentric nodal base gives results similar to those of the nodal base built from orthogonal bases, and they give more accurate results than does the Silvester nodal base. A small leveling off at the end for the barycentric modal base is due to an integration error for the high-order nodal base.

5.1.2 Derivative

The derivative errors for these scalar bases are shown on the right of Fig. 5. The results are similar to those for the interpolation errors. Since the accuracy of interpolation is better than that of the corresponding derivative errors, however, the small leveling off

Figure 5: Comparison of interpolation (left) and derivative (right)

Table 1: Relative integration errors using different bases

P	Nodal Base, Uniform	Nodal, Nonuniform	Modal, Uniform	Modal, Nonuniform
0	2.742e-2	2.742e-2	3.606e-2	3.606e-2
1	1.011e-2	1.022e-2	1.033e-2	1.033e-2
2	4.957e-4	4.430e-4	6.354e-4	4.522e-4
3	1.272e-4	8.000e-5	1.871e-4	1.091e-4
4	6.444e-6	3.432e-6	1.084e-5	4.789e-6
5	1.317e-6	4.666e-7	3.183e-6	1.064e-6
6	6.123e-8	1.978e-8	1.736e-7	5.482e-8

for the barycentric nodal base at P=6 for the derivative error is better than for the case of interpolation error. Similar to the interpolation error comparison, the differences of derivative between those three bases are small.

5.1.3 Integration

Next, we compare the integration error with two scalar bases. One is the nodal base built from orthogonal bases, and the other is the barycentric nodal base. The integration function is $f(x,y,z) = exp(x+y+z)$, and the number of total tetrahedron elements is 204. The computation domain is $[-\pi,\pi]^3$, and the accurate value for this integration is 12322.354615553. Two types of meshes have been used: uniform and nonuniform meshes. The nonuniform mesh uses the Fekete points on a tetrahedron element. As shown in Table. 1, the relative integration errors using nodal bases are better than using the modal base, and using nonuniform mesh gives better results than using the uniform mesh.

5.2 Vector Bases Comparison

Next, we investigate the vector bases. Similar to the scalar bases, interpolation, and derivative errors are discussed separately.

5.2.1 Interpolation

The interpolation errors with the vector bases are shown in Fig. 6. The total error is defined as $L_\infty(\vec{V})=\sqrt{L_\infty^2(V_x)+L_\infty^2(V_y)+L_\infty^2(V_z)}$. The vector function is $\vec{V}^T=(exp(y),exp(z),exp(x))$. For hierarchical vector bases, two meshes have been tested: uniform and nonuniform meshes. Therefore, three lines are shown in each plot of Fig. 6. One can see that the accuracies of these two vector bases are close to each other. The hierarchical vector base gives more accurate results than do the interpolatory vector bases. The error decreases more regularly for the interpolatory vector bases. The difference between uniform and nonuniform meshes is not prominent. The results are consistent with those of the scalar bases, where the Silvester nodal base gives a slightly worse result.

5.2.2 Derivative and Curl

The derivative and curl errors are shown in Fig. 7. The test function is $\vec{f}^T=(f_x,f_y,f_z),f_x=exp(y),f_y=exp(z),f_z=exp(x)$, and the number of total tetrahedron elements is 204. The computation domain is $[-1,1]^3$. Contrary to the interpolation, in both derivative and curl operations, the interpolatory vector base gives better results than does the hierarchical vector base. The reason could be the method of constructing vector bases, since a particular combination of different orders of barycentric coordinates was chosen. Detailed construction methods can be found in Appendix B. This characteristic is difficult to analyze in theory but easy to show through numerical comparison. Another characteristic is that the error curves using hierarchical vector bases are not smooth, whereas the curves using interpolatory vector bases is smooth.

5.3 Condition Number

Next, we investigate the condition number for these scalar and vector bases since it is directly related to the performance of the linear solver. Similar to above, only 3D bases are studied on a tetrahedron element.

5.3.1 Reference Element

We compare the condition numbers for the mass matrix on a tetrahedron element using nodal and modal scalar bases. In Table 2, the second and third columns show the condition numbers for using the nodal bases built from the orthogonal bases with uniform and nonuniform meshes, and the last column shows the condition numbers for using the barycentric modal base. Since the integration of bases product has analytical values, we just list the theoretical values for comparison. Clearly the condition numbers are much better when using the nodal base for both uniform and nonuniform meshes.

Figure 6: Comparison of interpolation (scalar: upper left, vector x component: upper right, vector y component: lower left, vector z component: lower right) on box domain

Figure 7: Comparison of derivative and curl operations on a box domain

Table 2: Mass matrix condition numbers using different bases for scalar bases

P	Nodal base, Uniform	Nodal, Nonuniform	Modal, theory
0	35.9752	35.9752	134.2906
1	105.0284	110.3535	2.0079e3
2	236.9213	250.7013	4.4464e4
3	409.8688	352.5142	1.1348e5
4	1.0888e3	814.0306	1.1769e7
5	2.7354e3	1.7491e3	2.5617e10
6	8.4537e3	4.4996e3	3.1260e13

Table 3: Mass matrix condition numbers using different bases for vector bases

P	Hierarchical, Uniform	Hierarchical, Nonuniform	Interpolatory, Uniform
0	10.0	10.0	10.0
1	696.3301	696.3300	79.8616
2	5.4905e5	3.5091e5	710.5062
3	3.0902e7	4.0057e7	2.1147e3
4	2.1910e9	4.5849e9	1.3570e4
5	8.9307e10	7.7603e13	5.3653e4
6	5.1629e13	1.7875e14	4.3495e5

Moreover, using a nonuniform mesh gives slightly better condition numbers. These results will influence the vector bases, since they are closely related to these two scalar bases.

Next, we compare the condition numbers for the mass matrix on a tetrahedron element using hierarchical and interpolatory vector bases. In Table. 3, the second and third columns show the condition numbers for using the hierarchical vector base with uniform and nonuniform meshes, and the last column shows the condition numbers for using the interpolatory vector base on a uniform mesh. As the interpolatory vector base is defined only on a uniform mesh, we just list one column for comparison. Clearly the condition numbers are much better when using the interpolatory vector base. Moreover, using a nonuniform mesh gives actually worse condition numbers than does the uniform mesh for the hierarchical vector base. These results are consistent with the scalar bases comparisons above. The results mean that the interpolatory vector bases have good independence and that the nonuniform mesh makes the independence of the hierarchical vector bases worse.

When we use the hierarchical vector base, we divide it into different categories connected to edge (4.3), face (4.4), and volume (4.5). In Table 4, their condition numbers are listed for comparison. The condition numbers for the edge mass matrix are relatively

Table 4: Mass matrix condition numbers using hierarchical vector bases

P	Edge, U	Edge, NU	Face, U	Face, NU	Volume, U	Volume, NU
0	1.0	1.0				
1	3.0	3.0	2.5652	2.6396		
2	25.6619	25.6619	2.8758e3	2.9106e3	3.6999	3.8622
3	237.6413	369.3742	6.1555e5	7.0088e5	1.6995e5	1.6247e5
4	3.2668e3	6.7008e3	8.5879e7	1.3755e8	6.6181e7	8.1105e7
5	9.1586e4	1.3863e5	3.6229e9	2.5345e10	4.6333e10	3.8163e10
6	2.2459e6	3.1862e6	2.1605e12	3.0497e12	9.6054e12	1.3160e13

Table 5: Condition numbers on box domain

P	0	1	2
A	1.0937e9	5.388e18/1.2185e10	1.1728e32/4.2682e11
B	7.8464	208.5681/71.8019	1.9168e7/1453.8874
A-4.0*B	184.6566	1630.0/1582.3	2.5135e5/7.6079e3
A-4.5*B	441.2363	2736.3 /3394.7	2.1197e5/1.7858e4
A-4.8*B	4957.8	16023.0/11023.0	6.3716e5/6.0083e4
B^{-1}*A	2.3811e9	2.8811e18/4.6730e10	3.3999e35/2.0014e12

small compared with those of the face and volume mass matrices. This means that the independence of the edge vector bases is relatively good, while the independence of the face and volume vector bases is not. The condition numbers of both face and volume mass matrices increase rapidly, and these are also consistent with the total mass matrix results above.

5.3.2 Global Domain

We now study the vector bases for the global domain $[-1,1]^3$. There are a total of 204 tetrahedron elements. Table 5 compares the condition numbers for the global mass and stiffness matrices. Some combinations of them have also been studied. For P=0, hierarchical and interpolatory vector bases are the same, so only one column of results is listed. For P=1 and 2, two columns of results are listed: the left one is for the hierarchical vector base, and the right one for the interpolatory vector base. The table clearly shows that using an interpolatory vector base gives better condition numbers for most cases, especially when P is large.

In Figure 8, the condition numbers are shown with polynomial order P. Clearly the condition numbers of the hierarchical vector base increase much rapidly than for the interpolatory vector base. The condition number of $B^{-1} * A$ is closer to the condition number of A than B. In Fig. 9, changes of the condition numbers with different λ have

Figure 8: Condition numbers with polynomial order: hierarchical (left) and interpolatory (right) vector bases

Figure 9: Condition numbers with shifted λ: hierarchical (left) and interpolatory (right) vector bases

Table 6: Comparison on degree of freedom (DOFs) (E=204)

Polynomial Order	CG	DG	Vector Base
3	537	4902	165
4	2094	10184	1038
5	6618	18310	3231

been plotted, and there are some values of λ that can achieve the smallest condition number for matrix $A-\lambda*B$. These values can be used in the eigenvalue computations to reach the fastest speed.

6 Applications

Now we investigate the performance of these two vector bases for some eigenvalue computations. First, we compare the degree of freedom (DOF) for scalar and vector bases on the same mesh in Table 6. For the scalar base, we consider continuous Galerkin (CG) and discontinuous Galerkin (DG) methods. As can be seen, the DOF of the DG method has the largest numbers, and vector bases are the least. The CG method has fewer DOFs than does the DG method. The difference between these different bases and methods becomes smaller as the polynomial order increases.

6.1 Simple Geometries

First, we test the two vector bases in a simple geometry, namely, a cubic box with size $[-1,1]^3$. Tables 7 and 8 give the smallest five eigenvalues computed by using the hierarchical and interpolatory vector bases, respectively, normalized with $\pi^2/4$. The theoretical values are given in the second column. Two different meshes have been used: one with 204 tetrahedra and the other with 55,713 tetrahedra. For the first mesh with 204 tetrahedra, three different polynomial orders were used. Results show that the accuracy increases when using the high-order bases. These results also show that both hierarchical and interpolatory vector bases produce the correct results. No spurious modes exist, as happens with scalar bases been reported in the literature.

6.2 Accuracy

To investigate the accuracy, we used a series of meshes shown in Fig. 10, varying by 8 times from 96 to 768 and from 768 to 6144, so the mesh size is halved for each pair of them. In Table 9, the average of the first three eigenvalue errors is listed for both vector bases. The left one is from interpolatory vector base, and the right one is from the hierarchical vector base.

Table 7: Eigenvalues on box domain using hierarchical vector bases

Mode	Theory	E=204,P=0	E=204, P=1	E=204, P=2	E=55k, P=0
1	2.0	1.9567	1.9798	1.9981	2.0001
2	2.0	1.9895	1.9839	1.9993	1.9997
3	2.0	2.004	1.9847	1.9993	1.9997
4	3.0	2.8904	2.9363	2.9902	2.9971
5	3.0	2.9894	2.9505	3.0190	2.9975

Table 8: Eigenvalues on box domain using nodal vector bases

Mode	Theory	E=204,P=0	E=204, P=1	E=204, P=2	E=55k, P=0
1	2.0	1.9567	2.0013	2.0001	2.0001
2	2.0	1.9895	1.9993	2.0001	1.9997
3	2.0	2.004	1.9989	2.0001	1.9997
4	3.0	2.8904	3.0080	3.0003	2.9971
5	3.0	2.9894	3.0064	3.0003	2.9975

Figure 10: Box mesh with 96, 768, and 6144 tetrahedra

Table 9: Eigenvalue errors with different mesh and polynomial orders

Element Number vs. P	0	1	2
96	2.1213E-1/2.1213E-1	8.2995E-3/3.2980E-2	3.5794E-4/4.1532E-3
768	3.5663E-2/3.5663E-2	6.2839E-4/1.8450E-3	6.2446E-6/6.7253E-5
6144	7.9908E-3/7.9908E-3	4.1159E-5/1.1178E-4	9.8685E-8/9.3207E-6

Figure 11: Comparisons on the average errors of first three eigenvalues

To compare the accuracy more clearly, we plot the results in Fig. 11. The left figure shows the average of the first three eigenvalue errors with different mesh sizes. The mesh size of 96 tetrahedra elements has been set to 1, and mesh sizes of 768 and 6,144 tetrahedra elements are then 0.5 and 0.25. Solid lines are the results of using interpolatory vector bases, and the dashed lines are from hierarchical vector bases. Since for P=0, the interpolatory and hierarchical vector bases are the same, only one has been plotted. From the left figure, the errors of using interpolatory vector base are smaller than using the hierarchical vector base. The difference of the errors for P=2 is larger than that for P=1. The right part of Fig. 11 shows the average of the first three eigenvalue errors with different polynomial orders. Similarly solid lines are results of using interpolatory vector bases, and the dashed lines are from hierarchical vector bases. For all three meshes, the errors decrease faster when using the interpolatory vector base. These comparisons clearly show that the interpolatory vector base can achieve more accurate results.

6.3 Speed

Although two vector bases give nearly the same correct results, the speed of the eigenvalue solvers is different. As shown in Table 10, when the polynomial order increases, the speed becomes slower for both bases. However, using the interpolatory vector bases is much faster than using the hierarchical vector bases. This result is consistent with the condition number effect above, since a large condition number slows the convergence rate when solving $A \cdot x = b$ in the eigenvalue solver.

Table 10: Speed comparison for eigenvalue solvers using 64 CPUs (seconds)

Bases	E=204,P=0	E=204, P=1	E=204, P=2	E=55k, P=0
Hierarchical	5.077123e-01	1.979933e+01	1.048209e+03	3.363760e+02
Interpolatory	5.191548e-01	1.579513e+01	1.520600e+02	3.321949e+02

Table 11: Scalablity comparison for eigenvalue solvers E=204, P=1 (seconds)

CPU	2	4	8
Hierarchical	9.056643e+01	6.727574e+01	3.677857e+01
Parallel Efficiency	1.0	0.673	0.615
Interpolatory	7.770730e+01	4.831998e+01	2.682341e+01
Parallel Efficiency	1.0	0.804	0.725

6.4 Scalability

Table 11 shows the scalability of two solvers for E=204, and P=1. When a small number of processors is used, the parallel efficiency is fine. For this situation, the interpolatory vector base gives better parallel efficiency. When a large number of processors is used, however, the parallel efficiency becomes worse, because of the iterative method used in the solvers. We plan to develop more scalable eigenvalue solvers in the future.

6.5 Complex Geometry

We applied also our eigenvalue solvers to simulate the Quater Wave Resonator (QWR) used in accelerator research. The mesh is shown in Fig. 12, with 8,502 tetrahedra in the simulation.

Table 12 shows the eigenvalues and their computation times for different solvers with different polynomial orders. The results are consistent. For example, the computation time for P=1 is nearly three times larger than that of P=0, consistent with the condition number studies above.

Figures 13 and 14 show the first resonating field in the (x, y), (x, z), and (y, z) planes. These results are from the solver using a hierarchical vector base with polynomial order 2.

Table 12: Comparison for complex geometry computation

P	0	1
Eigenvalue	2.7683/2.7683	2.7922/ 2.8369
Frequency (MHz)	79.44/79.44	79.78/ 80.42
Time (s)	311.68/288.60	944.66/951.61

Figure 12: QWR mesh E=8502

Figure 13: QWR resonating electric field in (x, y) (left) and (x, z) (right) planes

Since a perfect electric conduct boundary condition is applied on all boundaries, the electric field vectors are all perpendicular to the boundary. The most intensive field locations are on the axis and corner with small rounding radius. The field has three symmetries. The unsymmetric effects are due to the uneven element distribution.

7 Summary

This paper presented our efforts in studying the performance of eigenvalue solvers using two vector bases: hierarchical and interpolatory. The theories of these two bases have been explained briefly, and the construction methods have been discussed in detail. Domain decomposition has been used for parallelizing the solvers. Benchmarks of these parallel solvers have shown that the scaling is restricted by the difficulty of solving large sparse matrices with a large number of processors. Detailed comparisons have been performed based on numerical experiments. The accuracy, condition numbers, and uniform and nonuniform meshes have all been investigated. These results show that hierarchical scalar and vector bases usually have larger condition numbers for the mass matrix generated from them, leading to more iteration steps and slow speeds in the eigenvalue solvers. A nonuniform mesh sometimes can improve the accuracy of interpolation and derivative operators, but it may increase the condition number in some cases. Furthermore, two solvers have been compared for an eigenvalue computation on both simple and complex geometries, and the accuracy has been investigated based on mesh size and polynomial order. Although the accuracies become much better, the speeds of the eigenvalue solvers slow when high-order bases are used. In order to use high-order bases, new methods and techniques are needed.

Figure 14: QWR resonating electric field in (y, z) plane

ACKNOWLEDGEMENTS

This work was completed at ISCAS and ANL. This work was supported by the U.S. Department of Energy, Office of Nuclear Physics, under Contract No. DE-AC02-06CH11357. We thank the RDCPS support team at IS and the ALCF support team at ANL. Gail Pieper at MCS in ANL helped to proofread the paper.

Appendix A: Interpolatory Nédélec Vector Bases

Vector Basis on 2D Triangle Element

The zeroth-order vector bases function associated with the edge connecting nodes i_1 and i_2 is given by

$$\vec{\mathbf{W}}_{i_1 i_2} = \xi_{i_1} \nabla \xi_{i_2} - \xi_{i_2} \nabla \xi_{i_1}, \tag{7.1}$$

where ξ_i are barycentric coordinates related to nodes i, i=1,2,3. $\vec{\mathbf{W}}_{i_1 i_2}$ has only a tangential component along the edge connecting nodes i_1 and i_2.

In order to construct high-order vector bases, the interpolatory points should be chosen away from the triangle vertices. As shown on the left of Fig. 1, the interpolatory points are located only on the edges and face of the triangle element. The complete interpolatory polynomials are also associated with different edges as follows.

$$N_{ijk}^{i_1 i_2} = \alpha_{ijk}^{i_1 i_2} \frac{p+2}{\beta} \xi_{i_3} \hat{P}_i^{p+2}(\xi_1) \hat{P}_j^{p+2}(\xi_2) \hat{P}_k^{p+2}(\xi_3) \tag{7.2}$$

$$\hat{P}_j^{p+2}(\xi_i) = P_{j-1}^{p+2}(\xi_i - \frac{1}{p+2}) = \frac{1}{(j-1)!} \prod_{l=1}^{j-1} [(p+2)\xi_i - l], \tag{7.3}$$

where i_3 is an integer from (1,2,3) other than i_1 and i_2, and β is taken to be i,j, or k for i_3=1,2, or 3, respectively. For normalization, $\alpha_{ijk}^{i_1 i_2}$ has been chosen to make the component of $N_{ijk}^{i_1 i_2}$ along edge-connecting nodes i_1 and i_2 to be 1. One then finds that

$$\alpha_{ijk}^{i_1 i_2} = \frac{p+2}{p+2-\beta} \ell_{i_1 i_2}^{ijk}, \tag{7.4}$$

where $\ell_{i_1 i_2}^{ijk}$ is the length of the edge connecting nodes i_1 and i_2.

The total degree of freedom on triangle is (P+1)(P+3), where P is the order of the bases.

Vector Basis on 3D Tetrahedron Element

Similar to 2D case, the zeroth-order vector bases function associated with the edge connecting nodes i_1 and i_2 is given by

$$\vec{\mathbf{W}}_{i_1 i_2} = \xi_{i_1} \nabla \xi_{i_2} - \xi_{i_2} \nabla \xi_{i_1} \tag{7.5}$$

where ξ_i are barycentric coordinates related to nodes i, i=1,2,3,4. $\vec{\mathbf{W}}_{i_1 i_2}$ has only a tangential component along the edge conneting node i_1 and i_2.

In order to construct high-order vector bases, the interpolatory points should be chosen away from the tetrahedron vertices. As shown on the right of Fig. 1, the interpolatory points only located on the edges, faces, and volume of the tetrahedron element. The complete interpolatory polynomials are also associated with different edges as.

$$N_{ijkl}^{i_1 i_2} = \alpha_{ijkl}^{i_1 i_2} \frac{(p+2)^2}{\gamma\beta} \xi_{i_3} \xi_{i_4} \hat{P}_i^{p+2}(\xi_1) \hat{P}_j^{p+2}(\xi_2) \hat{P}_k^{p+2}(\xi_3) \hat{P}_l^{p+2}(\xi_4), \tag{7.6}$$

where i_3 and i_4 are two integers from (1,2,3,4) other than i_1 and i_2, and $\gamma(\beta)$ is taken to be i,j,k, or l for $i_3(i_4)$=1,2,3, or 4, respectively. $\hat{P}_j^{p+2}(\xi_i)$ is same as Eq. (7.3). For normalization, $\alpha_{ijkl}^{i_1 i_2}$ is same as $\alpha_{ijk}^{i_1 i_2}$ in Eq. (7.4) for the 2D case.

The total degree of freedom on the tetrahedron is (P+1)(P+3)(P+4)/2, where P is the order of the bases. More detailed information can be found in [11, 12, 15].

Appendix B: Hierachical Nédélec Vector Basis

Vector Basis on 2D Triangle Element

The zeroth-order vector bases function associated with the edge-connecting nodes i_1 and i_2 is given by

$$H^0(curl) = \xi_{i_1}\nabla\xi_{i_2} - \xi_{i_2}\nabla\xi_{i_1}, \tag{7.7}$$

where ξ_i are barycentric coordinates related to nodes i, i=1,2,3. $H^0(curl)$ has only a tangential component along the edge-connecting node i_1 and i_2.

In order to construct Pth-order vector bases, $H^p(curl)$, (P+1)th-order gradient type vector bases $\nabla W_{s,e}^{P+1}$ and $\nabla W_{s,f}^{P+1}$ should be added first. Then the (P+1)th-order nongradient type vector bases $\nabla W_{tv,e,ng}^{P+1}$ and $\nabla W_{tv,f,ng}^{P+1}$ should be added to make the vector bases complete to Pth-order,

$$H^p(curl) = W_{tv,e,ng}^{p} \bigoplus \nabla W_{s,e}^{p+1} \bigoplus W_{tv,f,ng}^{p+1} \bigoplus \nabla W_{s,f}^{p+1}, \tag{7.8}$$

where $W_{tv,e,ng}^{p} = W_{tv,e,ng}^{1} = H^0(curl)$. $W_{tv,f,ng}^{p+1}$ is the nongradient type vector bases associated with the face. $\nabla W_{s,e}^{p+1}$ and $\nabla W_{s,f}^{p+1}$ are gradients of corresponding potential bases associated with edges $W_{s,e}^{p+1}$ and face $W_{s,f}^{p+1}$. Their expressions are as follows

$$W_{s,e}^{p+1} = \{e_{1,2}^{p+1}, e_{1,3}^{p+1}, e_{2,3}^{p+1}, \quad (e_{m,n}^{p+1} = \lambda_m \lambda_n^i | i = 1,2,...,p)\} \tag{7.9}$$

$$W_{s,f}^{p+1} = \left\{ f_{1,2,3}^{p+1}, \quad \left(f_{1,2,3}^{p+1} = \lambda_1 \lambda_2^i \lambda^j \;\middle|\; \begin{matrix} i = 1,2,...,p-1 \\ j = 1,2,...,p-i \end{matrix} \right) \right\} \tag{7.10}$$

$$W_{tv,f,ng}^{p+1} = \{e_{2,3}^{p+1}\nabla\lambda_1, e_{1,3}^{p+1}\nabla\lambda_2, f_{1,2,3}^{p+1}\nabla\lambda_1\} \tag{7.11}$$

The total degree of freedom on a triangle is (P+1)(P+3), where P is the order of the bases. This is the same as for the interpolatory vector bases.

Vector Basis on 3D Tetrahedron Element

The zeroth-order vector bases functions associated with the edge-connecting nodes i_1 and i_2 is given by

$$H^0(curl) = \xi_{i_1}\nabla\xi_{i_2} - \xi_{i_2}\nabla\xi_{i_1}, \tag{7.12}$$

where ξ_i are barycentric coordinates related to nodes i, i=1,2,3,4. $H^0(curl)$ has only a tangential component along the edge-connecting nodes i_1 and i_2.

In order to construct Pth-order vector bases, $H^p(curl)$, (P+1)th-order gradient type vector bases $\nabla W_{s,e}^{P+1}$, $\nabla W_{s,f}^{P+1}$, and $\nabla W_{s,v}^{P+1}$ should be added first. Then the (P+1)th-order

nongradient-type vector bases $\nabla W^{P+1}_{tv,f,ng}$ and $\nabla W^{P+1}_{tv,v,ng}$ should be added to make the vector bases complete to Pth-order.

$$H^p(curl) = W^p_{tv,e,ng} \bigoplus \nabla W^{p+1}_{s,e} \bigoplus \nabla W^{p+1}_{s,f} \bigoplus \nabla W^{p+1}_{s,v} \bigoplus W^{p+1}_{tv,f,ng} \bigoplus W^{p+1}_{tv,v,ng}, \tag{7.13}$$

where $W^p_{tv,e,ng} = W^1_{tv,e,ng} = H^0(curl)$. $W^{p+1}_{tv,f,ng}$ and $W^{p+1}_{tv,v,ng}$ are the nongradient-type vector bases associated with faces and volume. $\nabla W^{p+1}_{s,e}$, $\nabla W^{p+1}_{s,f}$, and $\nabla W^{p+1}_{s,v}$ are gradients of corresponding potential bases associated with edges $W^{p+1}_{s,e}$, face$W^{p+1}_{s,f}$, volume $W^{p+1}_{s,e}$, and face $W^{p+1}_{s,v}$. Their expressions are as follows.

$$W^{p+1}_{s,e} = \{e^{p+1}_{1,2}, e^{p+1}_{1,3}, e^{p+1}_{2,3}, \quad (e^{p+1}_{m,n} = \lambda_m \lambda^i_n | i = 1,2,...,p)\} \tag{7.14}$$

$$W^{p+1}_{s,f} = \left\{ f^{p+1}_{1,2,3}, \quad \left(f^{p+1}_{1,2,3} = \lambda_1 \lambda^i_2 \lambda^j_3 \;\middle|\; \begin{array}{l} i = 1,2,...,p-1 \\ j = 1,2,...,p-i \end{array} \right) \right\} \tag{7.15}$$

$$W^{p+1}_{tv,f,ng} = \left\{ e^{p+1}_{l,n} \nabla \lambda_m, e^{p+1}_{n,m} \nabla \lambda_l, f^{p+1}_{l,m,n} \nabla \lambda_l \;\middle|\; (l,m,n) = \left\{ \begin{array}{l} (1,2,3),(1,3,4) \\ (1,2,4),(2,3,4) \end{array} \right\} \right\} \tag{7.16}$$

$$W^{p+1}_{tv,v,ng} = \left\{ \begin{array}{ll} f^{p+1}_{2,3,4} \nabla \lambda_1 & v^{p+1}_{1,2,3,4} \nabla \lambda_1 \\ f^{p+1}_{1,3,4} \nabla \lambda_2 & v^{p+1}_{1,2,3,4} \nabla \lambda_2 \\ f^{p+1}_{1,2,4} \nabla \lambda_3 & \end{array} \right\}, \tag{7.17}$$

where $f^{p+1}_{l,m,n}$ and $v^{p+1}_{1,2,3,4}$ are hierarchical potential bases that can be expressed as follows.

$$f^{p+1}_{l,m,n} = \left\{ \lambda_l \lambda^i_m \lambda^j_n \;\middle|\; \begin{array}{l} i = 1,2,\ldots,p-2, \\ j = 1,2,\ldots,p-1-i \end{array} \right\} \tag{7.18}$$

$$v^{p+1}_{1,2,3,4} = \left\{ \lambda_1 \lambda^i_2 \lambda^j_3 \lambda^k_4 \;\middle|\; \begin{array}{l} i = 1,2,\ldots,p-3, \\ j = 1,2,\ldots,p-2-i, \\ k = 1,2,\ldots,p-1-i-j \end{array} \right\} \tag{7.19}$$

The total degree of freedom on the tetrahedron is (P+1)(P+3)(P+4)/2, where P is the order of the bases. This is also same as for the interpolatory vector bases. More information can be found in [2, 35, 36, 39].

References

[1] A. Ahagon and T. Kashimoto, Three-dimensional electromagnetic wave analysis using high order edge element, *IEEE Trans. on Magnetics*, **31(3)**, 1753-1756, 1995.

[2] L.S. Andersen and J.L. Volakis, Hierarchical tangential vector finite elements for tetrahedra, *IEEE Trans. on Microwave Guided Wave Letters*, **8(3)**, 127-129, 1995.

[3] A. Bossavit, A rationale for edge-elements in 3-d fields computations. *IEEE Trans. on Magnetics*, **24(1)**, 74-79, 1988.

[4] A. Bossavit and I. Mayergoyz, Edge elements in scattering problems, *IEEE Trans. on Magnetics*, **25(4)**, 2816-2821, 1989.

[5] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang, Spectral Methods in Fluid Mechanics, Springer-Verlag, New York, 1987.

[6] Z. Cendes, Vector finite elements for electromagnetic field computation, *IEEE Trans. on Magnetics*, **27(5)**, 3958-3967, 1991.

[7] M.O. Deville, P.F. Fischer, and E.H. Mund, High-Order Methods for Incompressible Fluid Flow, Cambridge University Press, Cambridge, 2002.

[8] M. Dubiner, Spectral methods on triangles and other domains, *J. Sci. Comp.*, **6**, 345, 1991.

[9] I.S. Duff and J.K. Reid, The multifrontal solution of indefinite sparse symmetric linear equatiions, *ACM Transactions on Mathematical Software*, **8(3)**, 302-325, 1983.

[10] C. Geuzaine, B. Meys, P. Dular, and W. Legros, Convergence of high order curl-conforming finite elements, *IEEE Trans. on Magnetics*, **35(3)**, 1442-1444, 1999.

[11] R.D. Graglis, D.R. Wilton, and A.F. Peterson, Higher order interpolatory vector bases for computational electromagnetics, *IEEE Trans. on Antennas and Propagation*, **45(3)**, 329-342, 1997.

[12] R.D. Graglis, D.R. Wilton, A.F. Peterson, and I.L. Gheorma, Higher order interpolatory vector bases on prism elements, *IEEE Trans. on Antennas and Propagation*, **46(3)**, 442-450, 1998.

[13] A. Gupta, G. Karypis and V. Kumar, Highly scalable parallel algorithms for sparse matrix factorization, IEEE Trans. Parallel and Distributed Systems, **8**, 502-520, 1997.

[14] J.S. Hesthaven and T. Warburton, Nodal Discontinuous Galerkin Methods: Algorithms, Analysis and Applications, Springer, New York, 2008.

[15] J. M. Jin, The Finite Element Method in Electromagnetics, John Wiley & Sons. 1993.

[16] A. Kameari, Calculation of transient 3D eddy current using edge-elements, *IEEE Trans. on Magnetics*, **26**, 466-469, 1990.

[17] A. Kameari, Symmetric second order edge elements for triangles and tetrahedra, *IEEE Trans. on Magnetics*, **35(3)**, 1394-1397, 1999.

[18] G.E. Karniadakis and S.J. Sherwin, Spectral/hp element methods for CFD, Oxford University Press, London, 1999.

[19] J-F. Lee, D-K. Sun, and Z. Cendes, Full-wave analysis of dielectic waveguides using tangetial vector finite elemenets, *IEEE Trans. Microwave Theory and Techniques*, **39(8)**, 1262-1271, 1991.

[20] J-F. Lee, D-K. Sun, and Z. Cendes, Tangential vector finite elements for electromagnetic field computation, *IEEE Trans. on Magnetics*, **27(5)**, 4032-4035, 1991.

[21] R.B. Lehoucq, D.C. Sorensen and C. Yang, ARPACK Users' Guide: Solution of Large Scale Eigenvalue Probelms with Implicitly Restarted Arnoldi Method, Oct., 1997.

[22] X.S. Li and J.W. Demmel, SuperLU DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems, ACM Trans. Mathematical Software, **29(2)**, 110-140, 2003.

[23] J.W.L.Liu, The multifrontal method for sparse matrix solution: Theory and practice, SIAM Review, **34(1)**, 82-109, 1992.

[24] K.J. Mashhoff and D.C. Sorensen, PARPACK: An Efficient Portable Large Scale Eigenvalue Package for Distributed Memory Parallel Architectures, CRPC-TR96659, Oct., 1996.

[25] G. Mur and A. de Hoop, Edge elements, their advantages and their disadvantages, *IEEE Trans. on Magnetics*, **30(5)**, 3552-3557, 1994.

[26] J.C. Nédélec, Mixed finite elements in R^3, Numer. Math. **35**, 315-341, 1980.

[27] J.C. Nédélec, A new family of mixed finite elements in R^3, Numer. Math. **50**, 57-81, 1986.

[28] Peter Monk, Finite Element Methods for Maxwell's Equations, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, UK, 2008.
[29] J.S. Savage and A.F. Peterson, Higher-order vector finite elements for tetrahedral cells, *IEEE Trans. on Microwave Tech.*, **44(6)**, 874-879, 1996.
[30] Jie Shen and Tao Tang, Spectral and High-Order Methods with Applications, Science Press of China, 2006.
[31] P. Silvester, Finite Element Solution for Electrical Engineers, Cambridge University Press, Cambridge, UK. 1983.
[32] D. Sun, J. Manges, X. Yuan and Z. Cendes, Spurious modes in finite element methods, IEEE Antennas Propagation Magzine, **37(5)**, 12-24, 1995.
[33] J.L. Volakis, A. Chatterjee and L. Kempel, Finite Element Methods for Electromagnetics: Antennas, Microwave Circuits and Scattering Applications, IEEE Press, 1998.
[34] J.S. Wang and N. Ida, Curvilinear and higher order edge finite elements in electromagnetic field computation, *IEEE Trans. on Magnetics*, **29(2)**, 1491-1494, 1993.
[35] J.P. Webb and B. Forghani, Hierarchical scalar and vector tetrahedra, *IEEE Trans. on Magnetics*, **29(2)**, 1495-1498, 1993.
[36] J.P. Webb, Hierarchical vector basis functions of arbitrary order for triangular and tetrahedral finite elements, *IEEE Trans. on Antennas and Propagation*, **47(8)**, 1244-1253, 1999.
[37] H. Whitney, Geometric Integration Theory, Princeton University Press, 1957.
[38] T.V. Yioultsis and T.D. Tsiboukis, Development and implementation of second and third order vector finite elements in various 3-D electromagnetic filed problems, *IEEE Trans. on Magnetics*, **33(2)**, 1812-1815, 1997.
[39] Y. Zhu and A. Cangellaris, Multigrid Finite Element Methods for Electromagnetic Field Modeling, John Wiley & sons, Hoboken, New Jersey, 2006.